Merging syntactic lexica: the case for French verbs

Benoît Sagot, Laurence Danlos

Merging syntactic lexica: the case for French verbs

Benoît Sagot, Laurence Danlos

Alpage, INRIA Paris-Rocquencourt & Université Paris Diderot
175 rue du Chevaleret, 75013 Paris, France
benoit.sagot@inria.fr, laurence.danlos@linguist.jussieu.fr

Abstract

Syntactic lexicons, which associate each lexical entry with information such as valency, are crucial for several natural language processing tasks, such as parsing. However, because they contain a rich and complex information, they are very costly to develop. In this paper, we show how syntactic lexical resources can be merged, in order to take benefit from their respective strong points, and despite the disparities in the way they represent syntactic lexical information. We illustrate our methodology with the example of French verbs. We describe four large-coverage syntactic lexicons for this language, among which the Le*fff*, and show how we were able, using our merging algorithm, to extend and improve the Le*fff*.

Keywords: Lexicon Merging, Syntactic Lexicons, Le*fff*, Lexicon-Grammar, Dicovalence, LVF

1. Introduction

Syntactic lexicons are crucial for several natural language processing tasks, such as parsing, be it symbolic (Riezler et al., 2002; Thomasset and Éric de La Clergerie, 2005) or even statistical (Collins, 1997; Versley and Rehbein, 2009). Syntactic lexicons are rich and complex resources, and their development is a costly task. Although a lot of work has been published on the automatic acquisition of syntactic lexica, the resources that have a coverage and an accuracy large enough for being used as linguistic descriptions, e.g., in symbolic parsers, have been developed manually or semi-automatically, sometimes for several decades.

In this paper, we focus our study on French verbs. There exist today four large-coverage syntactic lexical resources for French, that provide information about the valency of lexical entries, i.e., subcategorization frames and other syntactic information relevant for describing the syntactic behaviour of predicative lexical units. These resources are Lexicon-Grammar tables (Gross, 1975; Boons et al., 1976b; Boons et al., 1976a; Guillet and Leclère, 1992), the verb valency lexicon Dicovalence (van den Eynde and Mertens, 2006), the verbal syntactico-semantic lexicon LVF (Dubois and Dubois-Charlier, 1997), and the Le*fff* (Sagot et al., 2006; Sagot, 2010). All these resources use both syntactic and semantic criteria for defining either one or several entries for the same verb lemma. Therefore, these lexicons can be considered as an inventory of *lexemes* (as opposed to verb lemmas) associated with syntactic information.

The objective of this paper is to show how these diverse resources can be leveraged for improving one of them, the Le*fff*, by developing and applying merging techniques for valency lexicons. In this paper, we limit ourselves to verbal entries, for at least two reasons. First, they are best covered in terms of syntactic information than other categories. For example, LVF and Dicovalence only cover verbs. Second, verb valency is crucial in the first NLP application of syntactic lexicons, namely parsing systems.

Merging syntactic lexicons is not a straightforward task. Indeed, there is no real consensus on the way syntactic information should be modeled and formalized. There are discrepancies among resources, which differ in various ways:

- coverage: for example, Dicovalence has focused on reasonably frequent entries of fairly frequent verbal lemmas, wheras LVF has tried to have as large a coverage as possible;
- level of granularity of the set of entries for a given lemma (i.e., level of granularity used for distinguishing lexemes from one another): for example, LVF can distinguish two entries which differ only at a very fine-grained semantic level, whereas other resources will contain only one corresponding entry (see examples below);
- nature and level of granularity of the syntactic properties they describe: for example, Lexicon-Grammar tables include a large amount of non-standard syntactic information (e.g., symetric verbs), but does not really cover reflexive and reciprocal realizations using the pronoun *se*, whereas Dicovalence only describes pronominal realizations of syntactic functions which include the reciprocal and reflexive *se* realizations,
- level of formalization: Dicovalence and the Le*fff* are immediately usable in NLP tools, contrarily to LVF or Lexicon-Grammar tables;
- definition of what is considered syntactic argument as opposed to an adjunct: Dicovalence considers as arguments complements that other resources sometimes consider as adjuncts.

The methodology we have developed for merging syntactic lexicons has been developed in the last years (Sagot and Danlos, 2008; Sagot and Fort, 2009; Sagot and Danlos, 2009; Molinero et al., 2009). Other teams have worked on this task, such as Crouch and King (2005) for English and Necşulescu et al. (2011) for Spanish. They address in different ways the issue of mapping lexical entries for

a given lemma from various input lexicons to the one another, although these entries might have been defined at least in part using semantic criteria. In the work by Crouch and King (2005), the authors rely on the fact that, in (some of) their input lexicons (VerbNet and Cyc), lexical entries, which correspond to lexemes, are associated with WordNet synsets. This allows them to put together lexical entries that are associated with identical or related senses, although they resort to non-trivial techniques for dealing with various types of discrepancies and inconsistencies. On the other hand, Necşulescu et al. (2011) simply want to merge subcategorization lexicons, i.e., lexicons that list all possible subcategorization frames for a given verb lemma (as opposed to lexeme). With their strategy, they avoid the need for correctly mapping to the one another lexical entries that are defined based on syntactico-semantic criteria. However, the resulting lexicon is then only a subcategorization lexicon, and not a full-featured syntactic lexicon associating syntactic information with each lexeme.
In our case, our input resources for French verbs do not contain WordNet synset information. Nevertheless, we do want to take advantage of sense distinctions between entries, and to produce a merged lexicon at the lexeme level, that preserves these sense distinctions to the appropriate extent. Our methdology can be sketched as follows. First, we chose a model for representing syntactic information, and convert all input resources in this model, after a careful linguistic analysis. In this paper, this common model is Alexina, the lexical framework on which the Le*fff* is based. This is because Alexina lexicons, as mentioned above, are immediately usable in NLP tools. Moreover, and contrarily to our other input lexicons, the Le*fff* strongly relies on the notion of syntactic function, which is the basis for many parsing systems. In a second step, we try and create *groupings*, i.e., sets of lexical entries possibly extracted from more than one input resources and that will be merged in one entry in the output lexicon. Finally, we perform the actual merging.
Such a methodology is useful for various reasons. Of course, it helps developing a resource that has a higher coverage and accuracy than all input resources, although some information might be lost during the conversion process. Second, it allows for an efficient manual work on the output resource, if such a work is considered; for example, pieces of information that originate in only one of the input resources are more dubious than others. Finally, as a consequence, it allows for detecting errors in the input lexicons, as will shall see below.
After a brief description of our four input syntactic lexicons in Section 2. illustrated with a running example, we describe in more details our merging methodology and algorithm (Section 3.). Then, we describe a set of experiments conducted in the last years that are based on this methodology (Section 4.). Finally, we draw several conclusions and indicate the next steps for this work.[1]

[1]If the paper is accepted, we will report on results we have obtained while trying to evaluate various syntactic lexicons by comparing the results of one of the best performing symbolic parsers for French when it uses one of these lexicons or another. These results are not included in this submision for space reasons.

2. Input resources

We shall not provide a detailed description of our input resources. Such descriptions can be found in the various publications related to each resource (see citations below). Rather, we shall illustrate these resources on a running example, the lemma *vérifier* 'check', 'verify'. In the reminder of this paper, we refer to the entry with id n for the lemma v in the resource i as v^i_n. For example, the (only) entry in the Le*fff* for the lemma *vérifier* is vérifier$^{\text{Le}fff}_1$. For simplification purposes, we use v^i_n both for the lexical entry in its original form and after its conversion in Alexina.

2.1. The Le*fff* and the Alexina lexical formalism

The Le*fff* (Lexique des formes fléchies du français — *Lexicon of French inflected form*) is a large-coverage syntactic lexicon for French (Sagot, 2010).[2] The current version of the Le*fff* (which is not the last one, as explained below) contains 10,214 entries for 7,813 distinct lemmas. Contrarily to the three other lexicons we have used, which were developped manually, the Le*fff* was developed in a semi-automatic way: automatic tools were used together with manual work (Sagot et al., 2006; Sagot, 2010).
The Le*fff* relies on the Alexina framework for the acquisition and modeling of morphological and syntactic lexicons. To represent lexical information, an Alexina lexicon relies on a two-level architecture:

- the *intensional* lexicon associates (among others) an inflection table and a canonical sub-categorization frame with each entry and lists all possible redistributions from this frame;
- the *compilation* of the intensional lexicon into an *extensional lexicon* builds different entries for each inflected form of the lemma and every possible redistribution.

The version of the Le*fff* that was available before the experiments described below (version 3.0b) contains only one entry for the lemma *vérifier*. Here is a simplified version of this entry:

vérifier$^{\text{Le}fff}_1$ *Lemma;v;<Suj:cln|sn, Obj:(cla|qcompl|scompl|sinf|sn)>; %ppp_employé_comme_adj,%actif,%passif, %se_moyen_impersonnel,%passif_impersonnel*

It describes a transitive verb whose arguments have the *syntactic functions* *Suj* and *Obj* listed between angle brackets, and which allows for the functional redistributions *past participle used as an adjective*, *active* (the default distribution), *impersonal middle-voice "se" construction*, *impersonal passive*, and *passive*.
The different syntactic functions are defined in the Le*fff* by criteria close to that used in Dicovalence, i.e., they rely for a large part on cliticization and other pronominal features. The Le*fff* uses the following syntactic functions: *Suj* (subject), *Obj* (direct object), *Objà* (indirect object canonically introduced by preposition "à"), *Objde* (indirect object canonically introduced by preposition "de"), *Loc*

[2]The Le*fff* is freely available under the LGPL-LR license. See http://gforge.inria.fr/projects/alexina/

(locative), *Dloc* (delocative), *Att* (attribute), *Obl* or *Obl2* (other oblique arguments).
Each syntactic function can be realized by three types of *realizations*: *clitic pronouns*, *direct phrases* (nominal phrase (*sn*), adjectival phrase (*sa*), infinitive phrase (*sinf*), completive (*scompl*), indirect interrogative (*qcompl*)) and *prepositional phrases* (direct phrases preceded by a preposition, such as *de-sn*, *à-sinf* or *pour-sa*). Finally, a function whose realization is not optional has its realizations list between angle brackets.[3]
The way morphological and syntactic information is encoded it the Le*fff* is such that the Le*fff* be directly used in NLP tools. For example, we are aware of several parsers using the Le*fff*, and based on various formalisms: LTAG, including LTAGs generated from meta-grammars developed in various meta-grammar formalisms (Thomasset and Éric de La Clergerie, 2005), LFG (Boullier and Sagot, 2005), and less well-known formalisms such as Interaction Grammars or Pre-Group Grammars.

2.2. Lexicon-Grammar tables

In the Lexicon-Grammar (Gross, 1975; Boons et al., 1976b; Boons et al., 1976a; Guillet and Leclère, 1992), the 14,000 entries for verb lexical entries are structured in the form of 61 *classes*, each class being described in a different *table*.[4] Each class (table) is defined by a *defining property*, which is valid for all lexical entries belonging to the class (i.e., the defining property described a sub-categorization that is valid for all entries in the class, although other sub-categorizations might be also valid for a given entry). Lexicon-Grammar tables include two entries for the lemma *vérifier*, which both belong to class 6. Let us illustrate the notion of defining property and the content of the corresponding Lexicon-Grammar table using these entries. The defining property for class 6 is N_0VQuP, which means that all entries in this class are transitive and may have a finite or infinitive clause as the realization of the second argument in addition to the default noun phrase realization. Note that the notion of syntactic function is absent from the Lexicon-Grammar model. In table 6, 40 additional properties are "coded", i.e., each entry specifies whether it has each property or not, in the form of a matrix with one entry per row and one property per column. Among these 40 properties (the set of properties differs from one table to another), we can cite for example $N_1 =: QuPind$ (if the second argument is a finite clause, its verbal head is at the indicative mood) or $N_1 =: N_{\mathrm{hum}}$ (its second argument can be human).
The two Lexicon-Grammar entries the lemma *vérifier* are associated (among others) with the properties shown below (including the defining property for class 6). However, the second entry is not yet coded: the only thing we know about this entry is that it satisfies the defining property. As for the first one, the properties we have indicated here show respectively that its periphrastic inflected froms are built using the auxiliary *avoir*, that the second argument (N_1) is not mandatory, that it can be realized (among others) as a finite clause whose verbal head is at the indicative mood, as a pre-verbal particle (a clitic pronoun) or as a non-human noun phrase, and finally that it can be passivized (the subject becoming a non-mandatory argument introduced by the preposition *par*).

vérifier$^{\mathrm{LG}}_{6\_504}$	*Aux =: avoir*
	N0 V
	N1 =: Qu Pind
	N1 =: Qu P = Ppv
	N1 =: N-hum
	[passif par]
	Ex: Max a vérifié que la porte était fermée
	'Max checked that the door was closed'
vérifier$^{\mathrm{LG}}_{6\_505}$	*(unknown, the lexical entry is not yet coded)*
	Ex: Les faits vérifient cette hypothèse
	'The facts validate this hypothesis'

2.3. Dicovalence

Dicovalence (van den Eynde and Mertens, 2006) is a verb valency lexicon for French that is a follow up to the PROTON lexicon.[5] It was developed in the Pronominal Approach framework (Blanche-Benveniste et al., 1984). In order to identify the valency of a predicate (i.e., its dependants and their properties), the Pronominal Approach uses the relation that exists between so-called *lexicalized* dependants (realized as syntagms) and pronouns that "intentionally cover" these possible lexicalizations. Pronouns (and "paranouns", cf. below), contrarily to syntagms, syntactic functions or thematic roles, have two important advantages: (1) they are purely linguistic units, and do not have any of the properties (e.g., semantic properties) that make grammaticality judgements about sentences with lexicalized dependants difficult to motivate; (2) there are only a limited amount of such units: their inventory is finite. Note that the pronouns used in Dicovalence are more numerous than what is usually called a pronoun. Indeed they also include what Dicovalence calls "paranouns", that differ from pronouns because they can be modified (as *rien* 'nothing' in *rien d'intéressant* 'nothing interesting') and because they can not be taken up by a syntagm (cf. **il ne trouve rien, les preuves* 'He finds nothing, the evidences', vs. *il les trouve, les preuves* 'He finds them, the evidences').
In Dicovalence, pronouns are grouped in *paradigms*, which correspond only approximately to syntactic functions (e.g., P0 corresponds to the subject, P1 to the direct object, and so on). But Dicovalence contains more paradigms than the usual inventories contain syntactic functions. For example, it licenses a quantity paradigm (PQ), a manner paradigm (PM) and others.
The version of Dicovalence used in the experiments described below[6] consists in a list of 8,214 entries for 3,729 unique verbal lemmas. These lemmas and entries are explicitely chosen because they are reasonably frequent.
Table 1 shows both (simplified) entries given for the

[3] Other information are encoded in the Le*fff*, such as control, mood for finite clause argument realizations, and others.

[4] Lexicon-Grammar tables are freely available under the LGPL-LR license. See http://ladl.univ-mlv.fr/.

[5] Dicovalence is freely available under the LGPL-LR license. See http://bach.arts.kuleuven.be/dicovalence/

[6] It is the version labeled 061117, which is not the last version. Experiments about the last version of Dicovalence are planned.

vérifier$^{DV}_{85770}$	
VAL$	vérifier: P0 (P1)
VTYPE$	predicator simple
EG$	je vérifierais cette information avant de la publier
P0$	qui, je, nous, elle, il, ils, on, celui-ci, ceux-ci
P1$	0, que, la, le, les, en Q, ça, ceci, celui-ci, ceux-ci, le(qpind), ça(qpind), le(qpsubj), ça(qpsubj), le(sipind), ça(sipind), le(indq), ça(indq)
RP$	passif être, se passif

vérifier$^{DV}_{85780}$	
VAL$	vérifier: P0 P1
VTYPE$	predicator simple
EG$	l'expérience a vérifié son hypothèse
P0$	que, elle, il, ils, ça, celui-ci, ceux-ci
P1$	que, la, le, les, en Q, ça
RP$	passif être, se passif

Table 1: Entries for *vérifier* in Dicovalence.

lemma *vérifier* in Dicovalence. These two entries exactly correspond to the two entries found in the Lexicon-Grammar: The example in entry 85770 means 'I will check this piece of information before I publish it', and the example in entry 85780 'The experiment validated his hypothesis'.

2.4. The *Lexique des Verbes Français*

The LVF (*Lexique des Verbes Français*) is a dictionary of French verbs developed by Dubois and Dubois-Charlier (Dubois and Dubois-Charlier, 1997) that has the form of a thesaurus of syntactico-semantic classes, i.e., semantic classes defined using syntactic criteria. It is a very large coverage resource, that gathers syntactic and semantic information. The different classes, that contain 25,610 entries, are defined by both syntactic and semantic features and form a three-level hierarchy. At the lowest level of the hierarchy, sub-sub-classes are either homogeneous in terms of syntactic behaviour, or are divided once more in sets of entries with entries that all have the same syntactic behaviour. However, these syntactic behaviours are coded in a compact but abstruse way, that we shall illustrate on our running example.

In LVF, *vérifier* has three distinct entries. Two of them are in the sub-sub-class P3b of transitive verbs "of the type 'target one's thinking activity towards something'". It belongs to the sub-class P3 for verbs expressing the 'manifestation of a thinking activity towards somebody or something', which is a sub-class of the larger class P of psychologic verbs. The last entry belongs to class D of verbs like *donner* 'give', sub-class D3 containing verbs with a figurative meaning "giving something to somebody" or "obtaining something from somebody", sub-sub-class D3c of verbs meaning "granting validity to something or value to somebody". These entries contain, among other things, the following information:

vérifier$^{\mathrm{LVF}}_{1}$	*P3b*	*T1400 P3000*
vérifier$^{\mathrm{LVF}}_{2}$	*P3b*	*T1300 P3000*
vérifier$^{\mathrm{LVF}}_{3}$	*D3c*	*T3300*

The third column contains the syntactic codes. For example, code T1400 indicates a transitive construction with a human subject and a non-human (nominal) or clausal direct object. Code P3000 a pronominal construction with a non-human subject. T3300 stands for a transitive construction with non-human nominal subject and object, whereas T3100 stands for a transitive construction with a non-human nominal subject and a human object. On the one hand, these examples show, although not very clearly, a general fact: syntactic descriptions in LVF are less fine-grained than those found in other resources, except for semantic properties of the arguments, in particular prepositional ones. On the other hand, one can see that the inventory of lexical entries is more fine-grained than in other resources: the first two entries introduce a distinction that is not present in Dicovalence or in Lexicon-Grammar tables, which puts them together in only one entry (*vérifier*$^{DV}_{85770}$ and *vérifier*$^{\mathrm{LG}}_{6\_504}$). The third entry directly matches entries *vérifier*$^{DV}_{85780}$ and *vérifier*$^{\mathrm{LG}}_{6\_505}$).

3. Merging algorithm

As sketched in the Introduction, our merging algorithm is a three-step process (Sagot and Danlos, 2008):

- converting all input resources into the common model, which, as explained above, is Alexina; all converted resources must use the same inventory of syntactic functions, realizations and redistributions — in our case, that of the L*efff*; the main challenge at this stage is to be able to extract as much information as possible from the input resources and encode them in the form of an Alexina lexicon, despite all discrepancies between resources, as seen in the previous section;
- creating *clusters* of entries from various resources such that the entries in each should be merged into one entry; this step is very challenging, as its aim is to address the discrepancies in the granularity of lexical entries from one lexicon to another; for example, it is reasonable to consider that entries *vérifier*$^{DV}_{85770}$, *vérifier*$^{\mathrm{LG}}_{6\_504}$ and both *vérifier*$^{\mathrm{LVF}}_{1}$ and *vérifier*$^{\mathrm{LVF}}_{2}$ for a unique grouping
- merging of these clusters into output lexical entries.

3.1. Converting input lexicon in the L*efff* format

The way the lexical information is structured in the L*efff* is not very different from what can be found in **Dicovalence**. This makes the conversion process for Dicovalence reasonably straightforward. It is based on the following principles, which are obviously approximations:

- each Dicovalence *paradigm* is mapped into a L*efff syntactic function*;[7]

[7] We insist on the fact that this is an approximation.

- each pronoun (or paranoun) in the Dicovalence paradigm is mapped into a L*efff* realization: for example, if the pronoun *te* belongs to paradigm *P1*, a realization *cla* (accusative clitic) is added to the syntactic function *Obj* (we lose here the fact that the direct object can be human);
- each Dicovalence reformulation is converted into a L*efff* redistribution.

Converting **Lexicon-Grammar tables** into the Alexina format is much more complex a task. Although the extraction of an NLP-oriented lexicon from Lexicon-Grammar tables has raised interest for some time (Hathout and Namer, 1998; Gardent et al., 2005), the only attempt that was successful in producing and using in a parser an NLP-lexicon from *all* lexicon-grammar tables is the work by Tolone and colleagues (Tolone and Sagot, 2011). The final output of this conversion process is an Alexina lexicon that is consistent with the L*efff* in terms of syntactic function, realization and redistribution inventories, and in terms of linguistic modeling. It is what we shall call the "full" Alexina version of Lexicon-Grammar tables.

Because this conversion process is complex, and before its results were available, we have also directly extracted a **"light" Alexina version of Lexicon-Grammar tables**, in the context of the work about pronominal constructions described in Section 4.1. (Sagot and Danlos, 2009). For each entry we have retained for this work, we only extracted its functional sub-categorization frame, i.e., the list of syntactic functions without their possible realizations, as well as some redistributions (active, passive and *se*-middle).

Building an **Alexina version of LVF** was simply achieved by parsing valency data (codes such as T3100) and generating on-the-fly the corresponding Alexina entries. The only pieces of information that required a few heuristics are syntactic functions, which are not all explicitely recoverable from LVF codes, but that can be inferred using LVF information of argument introducters (e.g., prepositions) and semantic types.

In the remained of this paper, E_n^i is the lexical entry n in the lexicon i after it is converted in the Alexina format. Thus, i can be Dicovalence, LVF or L*efff*, as well as LG for the "full" Alexina variant of Lexicon-Grammar tables, and LG-light for the "light" version.

3.2. Grouping entries from various lexicons

For a given lemma, each resource might contain more than one entry. Therefore, it is necessary to determine the number of entries in the output merged lexicon, each of them being obtained by merging together one or several entries from each input lexicon, according to an algorithm detailed below. This means that the first step before merging is to build sets of entries for each lemma, each set corresponding to one output entry. We shall call such sets *groupings*.

Building such groupings is a challenging task. Indeed, what we need here is to identify cross-resource correspondances between entries, that are not necessarily one-to-one. Moreover, we can only rely on the information that is available in the input resources, i.e., mainly syntactic information, whereas distinctions between entries are often, at least in part, semantic. On the other hand, we have seen, while describing our input resources, and in particular the example of the entries for *vérifier*, that these resources have various granularities. In the case of *vérifier*, we can see that LVF entries are more fine-grained than Lexicon-Grammar and Dicovalence entries, which in turn are more fine-grained than the unique entry in the L*efff*. It turns out that this ordering of the resources is the same for most verbal lemmas. Therefore, we have chosen to base our grouping algorithm on an *inclusion* relation, which formalizes this intuition.

We first define this *inclusion* relation at the entry level as follows. For a given lemma, an entry E_1 is *included in* or *more specific than* an entry E_2 if and only if the set of clauses headed by an occurrence of entry E_1 is included in the set of clauses headed by E_2. Such an inclusion is noted $E_1 \subset E_2$. For example, $\textit{vérifier}_{85770}^{DV} \subset \textit{vérifier}_{1}^{Lefff}$.

Then, we generalize this inclusion relation at the resource level. Contrarily to the inclusion relation at the entry level, which can be defined without any problem, assuming that we can define an inclusion relation at the resource level is obviously an approximation, but it is required for being able to create these mappings. A lexicon i is considered included in or more specific than another lexicon j if we make the assumption that entries from lexicon i are *all* more specific than entries from lexicon j, i.e., each entry E_n^i is more specific than an entry E_m^j, except if the lexicon j does not contain any entry corresponding to E_n^i (in that case, i's coverage is higher than j's); assuming that i is more specific than j, this means that building groupings can be achieved by computing inclusion relations of the form $E_n^i \subset E_m^j$. Generalizing the example given above, we can posit that DV $\subset$ L*efff*.

We compute such inclusion relations using the following heuristics: starting from the resource-level hypothesis that $i \subset j$, an entry E_n^i is considered included in another entry E_m^j if it has exactly the same set of arguments with one of the base syntactic functions (subject, direct object, indirect object introduced by *à*, indirect object introduced by *de*) and if the set of syntactic functions of remaining arguments in E_m^j is included in that of E_n^i. One can see that we only rely on the inventories of syntactic functions. Moreover, we consider that only base synctactic functions can be used as safe clues, and that more oblique ones are likely to be found only in the most specific resource — but if one is found in the least specific resource, then it has to be in the specific one. In our case, this algorithm satisfyingly computes the relation $\textit{vérifier}_{85770}^{DV} \subset \textit{vérifier}_{1}^{Lefff}$, as they both entail, after conversion in the Alexina format, the syntactic functions *Suj* and *Obj* (the Alexina version of $\textit{vérifier}_{85770}^{DV}$ is shown below).

The groupings are then build as follows: we start from each entry that includes no other entry (often, an entry from the most specific lexicon) and we follow all inclusion relations until we reach entries that are include in no other entries. The set of all entries that are reached constitute a grouping. Of course, a grouping might end up containing entries from one resource only, if it is not included in any entry from another lexicon. If this is because this entry

corresponds to a meaning or a valency that is not covered by other resources, this is the expected result. However, it might be the result of mismatches resulting from the original resources, either because of errors in an input lexicon or because there are differences in the way a same construction is analyzed (e.g., a resource might consider as an indirect object in *à* what another resource analyzes as a locative argument). These problems, as well as the fact that entries might be incomplete (see the case of *vérifier*$^{\mathrm{LG}}_{6\_505}$), might also provoke erroneous groupings.

But this algorithm could still be improved. First, it can never put two distinct entries from the same lexicon in a same grouping. Going back to our running example, this algorithm would cluster all entries for *vérifier* in three groupings, each of them containing one of the three LVF entries, although we might wish only two. Second, restricting the information used for creating groupings to the inventory of syntactic functions is not always precise enough. In our running example, a correct mapping between LVF and Dicovalence entries for *vérifier* would require using the information about the human vs. non-human features applied to the subject. These improvements will be implemented in the future.

3.3. Merging entries

Once the groupings are built, we merge the entries in each grouping in a relatively straightforward way:

- the set of syntactic functions is built as the union of the set of syntactic functions in the input entries;
- for each syntactic function, the set of realizations is also obtained by union; for each realization we indicate which sources include it (no indication is added if it is licensed by all entries in the grouping);
- a realization is considered mandatorily realized only if it is mandatory in all entries in the grouping,
- the set of possible redistributions is built as the union of all possible redistributions in all entries in the grouping.

Let us illustrate the output of the merging of a grouping containing the entries *vérifier*$^{\mathrm{Dicovalence}}_{85770}$ and *vérifier*$^{\mathrm{Lefff}}_{1}$. The Le*fff* entry has been shown above. The Dicovalence entry in its original format was also given. Once converted in the Alexina model, it has become:

vérifier$^{\mathrm{DV}}_{85770}$ *Lemma;v;<Suj:cln|sn, Obj:(sn|cla|scompl|qcompl)>; %actif,%passif,%se_moyen*

Applying the merging algorithm leads to the following entry:

vérifier$^{\mathrm{Lefff}}_{1}$ +$^{\mathrm{DV}}_{85770}$ *Lemma;v;<Suj:cln|sn, Obj:(cla|qcompl|scompl|sinf*$^{\mathrm{Lefff}}$*|sn)>; %ppp_employé_comme_adj,%actif, %passif_impersonnel,%passif, %se_moyen,%se_moyen_impersonnel*

Note that the infinitive relalization of the direct object only comes from the Le*fff*, and is marked as such. This allows for a more efficient manual validation, if required, as a piece of information that is only licensed by one resource is more dubious than others. In this case, it is valid.

4. Merging experiments

4.1. Improving the coverage of the Le*fff* on pronominal entries with Dicovalence and Lexicon-Grammar tables

In order to improve the coverage of the Le*fff* over pronominal entries and pronominal constructions (i.e., realizations using the reflexive or the reciprocal *se*), we have leveraged the syntactic information Dicovalence, and to a lesser extend, Lexicon-Grammar tables, under the "light" version described above (Sagot and Danlos, 2009). First, we have carefuly described such constructions, and explored the way they were encoded in Dicovalence and in Lexicon-Grammar tables, as well as the way they were to be formalized in the Le*fff*. Then, we have extracted from Dicovalence and converted in the Alexina formalism the 5,273 entries that are either pronominal or that include realizations in *se*. Moreover, we have extracted 550 such entries using the "light" conversion scheme. We have merged the Le*fff* as well as these two additional sets of entries, using the inclusion relations DV $\subset$ Le*fff* $\subset$ LG-light. The result of the merging, which has since then be included in the Le*fff*, consists in 5,464 lexical entries.

4.2. Merging the Le*fff*, Dicovalence and LVF entries for denominal and deadjectival verbs in *-iser* and *-ifier*

In French, verbs in *-iser* and *-ifier* are particularly interesting. First, most of them are denominal or deadjectival verbs, which means they are relevant for studying the relation between morphological derivation and valency. Second, a large amount of verbal neologisms are built using one of these two morphological derivation mechanisms, and studying verbs in *-iser* and *-ifier* is an important step towards the development of tools for turning a syntactic lexicon into a *dynamic* lexicon that evolves in parallel with textual corpora.

Our work (Sagot and Fort, 2009) was based on the Le*fff*, Dicovalence and on LVF, which has a very large coverage. We have restricted it to verbs in *-iser* or *-ifier* that are indeed denominal or deadjectival, by manually removing other verbs ending "accidentally" in *-iser* or *-ifier*, such as *croiser* 'cross'. We relied on the following inclusion relations: LVF $\subset$ DV $\subset$ Le*fff*. The merging process created 2,246 entries covering 1,701 distinct lemmas (1,862 entries for verbs in *-iser* covering 1,457 distinct lemmas, and 384 entries for verbs in *-ifier* covering 244 distinct lemmas.

Note that this work was complemented with a corpus-based extraction step for finding missing denominal and deadjectival entries in *-iser* and *-ifier*.

4.3. Merging the Le*fff* and Dicovalence for increasing the granularity and the accuracy of the Le*fff*

In order, again, to increase the granularity and the accuracy of the Le*fff*, we have conducted a work aiming at merging the whole verbal lexicon of the Le*fff* and Dicovalence, and then validate or correct and/or merge manually the resulting entries. We have applied the methodology described in this paper using the inclusion relation DV $\subset$ Le*fff*. The, we have validated the 100 most frequent lemmas as well

as all *dubious* lemma, i.e., those lemma who got more entries in the merged lexicon than originally in both input lexicons. This validation step allowed us to remove erroneous realizations that were present in the Le*fff*, to indeed extend its coverage, accuracy, and fine-grainedness, but also to unvail errors in Dicovalence itself. This illustrates what we have explained above: not only merging syntactic lexicons lead to a improved output resource, but it also allows to improve the input resources themselves.

This work extended the number of verbal entries in the Le*fff* from 10,214 to 12,610, whereas the number of distinct lemmas was extended from 7,813 to 7,990 lemmas. The new version of the Le*fff* resulting from this automatic merging and manual validation and correction step is already freely available in the last distribution of the Le*fff*, but is not yet considered validated enough to replace the previous verbal lexicon files, which are therefore still distributed as well. It corresponds to the NewLe*fff* in the parsing evaluation work described in (Tolone et al., 2012).

5. Conclusion and next steps

In this paper, we have shown how syntactic lexical resources can be merged, in order to take benefit from their respective strong points, and despite the differences in the way they represent syntactic lexical information. We have described four large-coverage syntactic (including valency) lexicons for French, among which the Le*fff*, and have shown how we have used our merging algorithm for extending and improving the Le*fff*. In two experiments, we have merged up to 3 resources but restricting ourselves to two classes of entries. In the last experiments, all entries of only two lexicons were merged. Moreover, we used one of our input lexicons, namely Lexicon-Grammar tables, only in a light way, as explained below.

The next step of our work will be twofold. First, we will implement improvements that will address the limits of our grouping algorithm, as explained above. Second, we will finally merge our four lexical resources, including Lexicon-Grammar tables fully converted in the Alexina format (as opposed to the "light" version). This should give birth to a new version of the Le*fff*, which will then become the syntactic resource with the largest coverage, and hopefully a very high accuracy, concerning French verbs.

6. References

Claire Blanche-Benveniste, José Delofeu, Jean Stefanini, and Karel van den Eynde. 1984. *Pronom et syntaxe. L'approche pronominale et son application au français*. SELAF, Paris.

Jean-Pierre Boons, Alain Guillet, and Christian Leclère. 1976a. *La structure des phrases simples en français : Constructions intransitives*. Droz, Genève, Suisse.

Jean-Pierre Boons, Alain Guillet, and Christian Leclère. 1976b. La structure des phrases simples en français, classes de constructions transitives. Technical report, LADL, CNRS, Paris 7.

Pierre Boullier and Benoît Sagot. 2005. Efficient and robust LFG parsing: SXLFG. In *Proceedings of IWPT 2005*, pages 1–10, Vancouver, Canada.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 16–23, Stroudsburg, PA, USA.

Dick Crouch and Tracy Holloway King. 2005. Unifying Lexical Resources. In *Proceedings of the Interdisciplinary Workshop on the Identification and Representation of Verb Features and Verb Classes*, Saarbrücken, Germany.

Jean Dubois and Françoise Dubois-Charlier. 1997. *Les verbes français*. Larousse-Bordas, Paris, France.

Claire Gardent, Bruno Guillaume, Guy Perrier, and Ingrid Falk. 2005. Maurice Gross' Grammar Lexicon and Natural Language Processing. In *Proceedings of the 2nd Language and Technology Conference (LTC'05)*, Poznań, Pologne.

Maurice Gross. 1975. *Méthodes en syntaxe : Régimes des constructions complétives*. Hermann, Paris, France.

Alain Guillet and Christian Leclère. 1992. *La structure des phrases simples en français : Les constructions transitives locatives*. Droz, Genève.

Nabil Hathout and Fiammetta Namer. 1998. Automatic construction and validation of French large lexical resources: Reuse of verb theoretical linguistic descriptions. In *Proceedings of the 1st Language Resources and Evaluation Conference (LREC'98)*, Granada, Spain.

Miguel Ángel Molinero, Benoît Sagot, and Lionel Nicolas. 2009. A morphological and syntactic wide-coverage lexicon for spanish: The Le*ff*e. In *Proceedings of RANLP 2009*, Borovets, Bulgaria.

Silvia Necşulescu, Núria Bel, Muntsa Padró, Montserrat Marimon, and Eva Revilla. 2011. Towards the automatic merging of language resources. In *Proceedings of WoLeR 2011, the 1st International Workshop on Language Resources*, Ljubljana, Slovenia.

Stefan Riezler, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell, III, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 271–278, Stroudsburg, PA, USA.

Benoît Sagot and Laurence Danlos. 2008. Méthodologie lexicographique de constitution d'un lexique syntaxique de référence pour le français. In *Actes du colloque Lexicographie et informatique: bilan et perspectives*, Nancy, France.

Benoît Sagot and Laurence Danlos. 2009. Constructions pronominales dans dicovalence et le lexique-grammaire — intégration dans le lefff. *Linguisticæ Investigationes*, 32(2).

Benoît Sagot and Karën Fort. 2009. Description et analyse des verbes désadjectivaux et dénominaux en *-ifier* et *-iser*. In *Proceedings of the 28th Lexis and Grammar Conference*, Bergen, Norway.

Benoît Sagot, Lionel Clément, Éric de La Clergerie, and Pierre Boullier. 2006. The Le*fff* 2 syntactic lexicon for

French: architecture, acquisition, use. In *Proceedings of the 5th Language Resource and Evaluation Conference*, Lisbon, Portugal.

Benoît Sagot. 2010. The Le*fff*, a freely available, accurate and large-coverage lexicon for French. In *Proc. of the 7th Language Resource and Evaluation Conference*, Valetta, Malta.

François Thomasset and Éric de La Clergerie. 2005. Comment obtenir plus des méta-grammaires. In *Proceedings of TALN'05*, Dourdan, France, June.

Elsa Tolone and Benoît Sagot. 2011. Using Lexicon-Grammar tables for French verbs in a large-coverage parser. In Zygmunt Vetulani, editor, *Human Language Technology, Forth Language and Technology Conference, LTC 2009, Poznań, Poland, November 2009, Revised Selected Papers*, Lecture Notes in Artificial Intelligence (LNAI). Springer Verlag.

Elsa Tolone, Éric Villemonte de La Clergerie, and Benoît Sagot. 2012. Evaluating and improving syntactic lexica by plugging them within a parser. In *Proceedings of the 8th Language Resources and Evaluation Conference (LREC'12)*, Istanbul, Turkey. To appear.

Karel van den Eynde and Piet Mertens. 2006. Valency dictionary - DICOVALENCY: user's guide. See http://bach.arts.kuleuven.be/dicovalence/.

Yannick Versley and Ines Rehbein. 2009. Scalable discriminative parsing for German. In *Proceedings of the 11th International Conference on Parsing Technologies*, pages 134–137, Stroudsburg, PA, USA.